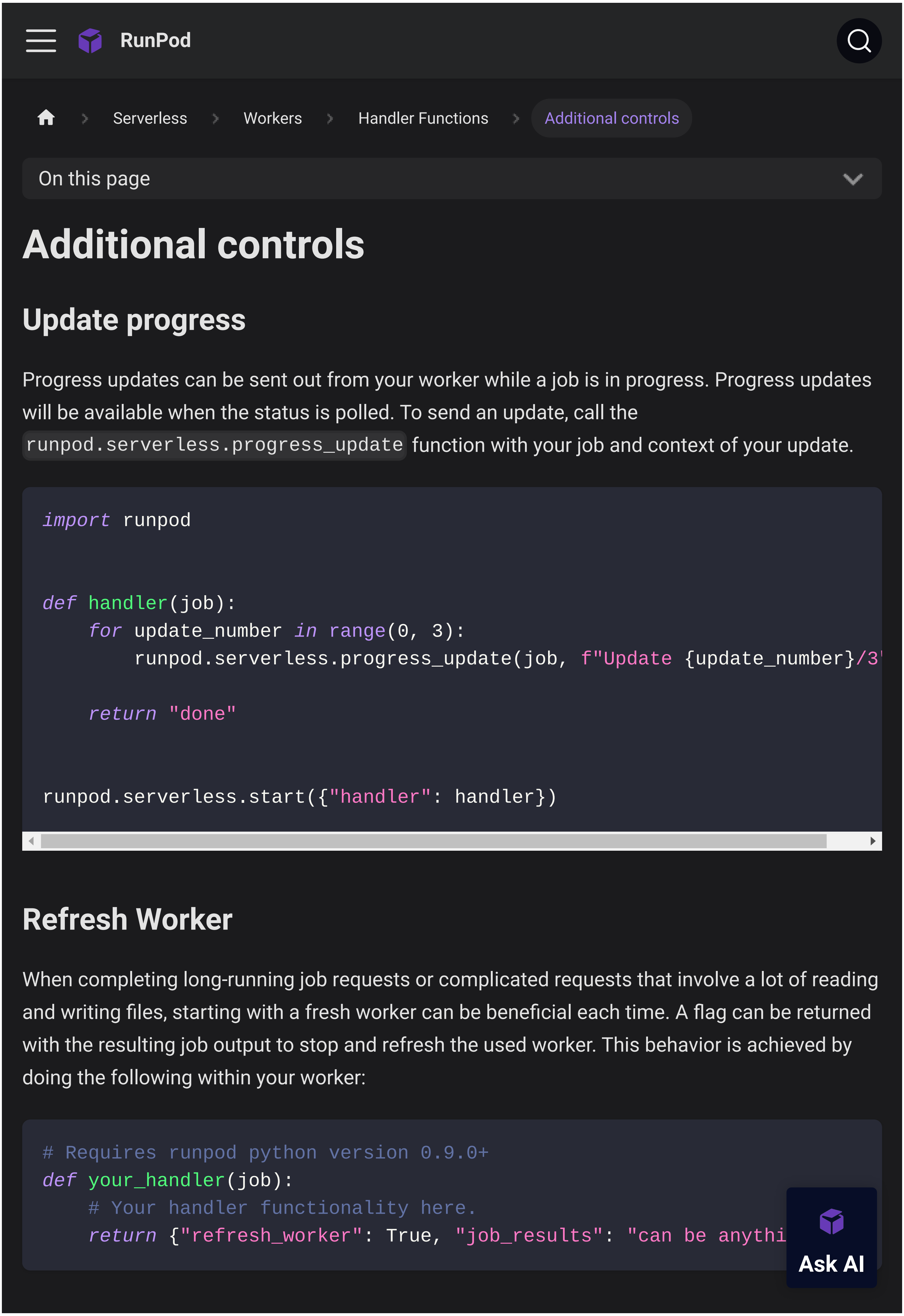

Additional controls

Update progress

Progress updates can be sent out from your worker while a job is in progress. Progress updates will be available when the status is polled. To send an update, call the `runpod.serverless.progress_update` function with your job and context of your update.

```
import runpod


def handler(job):
    for update_number in range(0, 3):
        runpod.serverless.progress_update(job, f"Update {update_number}/3

    return "done"


runpod.serverless.start({"handler": handler})
```

Refresh Worker

When completing long-running job requests or complicated requests that involve a lot of reading and writing files, starting with a fresh worker can be beneficial each time. A flag can be returned with the resulting job output to stop and refresh the used worker. This behavior is achieved by doing the following within your worker:

```
# Requires runpod python version 0.9.0+
def your_handler(job):
    # Your handler functionality here.
    return {"refresh_worker": True, "job_results": "can be anythi
```

Your handler must return a dictionary that contains the refresh_worker; this flag will be removed before the remaining job output is returned.

> ⓘ **NOTE**
>
> Refreshing a worker does not impact billing or count for/against your min, max, and warmed workers. It simply "resets" that worker at the end of a job.

Edit this page

Previous
« Handling Errors

Next
Development »